

Attorney Docket No: IDF 1540 (4000-02000)

Patent

AUG 07 2006

REMARKS

This application has been carefully considered in connection with the Examiner's Final Office Action dated June 19, 2006. Reconsideration and allowance are respectfully requested in view of the following.

Summary of Rejections

Claims 32-34 were pending at the time of the Final Office Action.

Claims 32-34 were rejected under 35 USC 103(a) as being unpatentable over Chiang et al. (U.S. Patent No. 6,948,174).

Summary of Response

Claim 32 was amended.

Claims 32-34 remain as previously presented.

New Claims 35-47 were added. These claims represent the same subject matter as original dependent Claims 3, 5-12, and 22-25. Applicants respectfully submit that these claims do not introduce any new subject matter, and their entry is respectfully requested.

Remarks and Arguments are provided below.

Summary of Claims Pending

Claims 32-34 are currently pending following this response.

Attorney Docket No: IDF 1540 (4000-02000)

Patent

Interview Summary

A telephone interview was conducted with Examiner Douglas Blair and Elizabeth Pham on August 2, 2006. Applicants would like to thank the Examiner for his time and consideration of this matter. Claim 32 was discussed in view of Chiang et al. (U.S. Patent No. 6,948,174). The Examiner noted that Chiang et al. did not appear to disclose a middleware brokering server as taught by the present application. The Examiner suggested that Claim 32 would be allowable unless a reference is found that teaches a middleware brokering server as disclosed by the present application.

Response to Rejections under Section 103

In the Final Office Action dated June 19, 2006, Claims 32-34 were rejected under 35 USC § 103(a) as being unpatentable over Chiang et al. (U.S. Patent No. 6,948,174).

With regard to independent Claim 32, this claim recites, "mapping the message in Cobol copybook format onto the fields in a structured event format."

The Office Action appears to suggest that Col. 10, lines 18-45 of Chiang et al. teaches mapping a message in Cobol copybook format onto the fields in a structured event format. However, Applicants are unable to find such a teaching in the cited section of Chiang et al. That section appears only to disclose parsing a Cobol source file into an XMI instance file. An XMI instance file is not the same as a structured event format. Having the message in a standard format allows the middleware broker server to operate in the publish/subscribe mode rather than the point-to-point mode thus minimizing the number of independent adapters needed for communication between the disparate middleware products. The standard format also reduces vendor dependency.

Attorney Docket No: IDI 1540 (4000-02000)

Patent

Claim 32 also recites, "communicating the message converted from Cobol copybook format to structured event format to a middleware brokering system."

The Office Action has suggested that Col. 10, lines 46-61 of Chiang et al. teaches communicating the message converted from Cobol copybook format to structured event format to a middleware brokering system. However, Applicants are unable to find such a teaching in the cited section of Chiang et al. This section merely states that middleware 713 invokes applications 703 through the application interfaces 705. A middleware invoking the applications that it serves through their application interfaces does not teach or suggest communicating a message to a middleware brokering system as disclosed by the present application.

Claim 32 also recites, "mapping the message in the JMS format onto the fields in a structured event format."

Again, the Office Action appears to suggest that Col. 10, lines 18-45 of Chiang et al. teaches mapping a message in JMS format onto the fields in a structured event format. However, Applicants are unable to find such a teaching in the cited section of Chiang et al. As established above, that section appears only to disclose parsing Java code into an XMI instance file. An XMI instance file is not the same as a structured event format.

Also, Chiang et al. discloses only the use of a Java application layer in the following passage:

"The Common Application Metamodel tool, method, and system is especially useful for providing a data transformer that is bi-directional between a client application and a server application, transmitting commands and data both ways between, for example, a Java, HTML, XML, C, or C++ application and a COBOL, PL/I, or High Level Assembler application, or, between an HTML or XML application and a Java, C, or C++ application, or between a Java application and a C or C++ application." (Col. 3, ll 42-52)

By contrast, the presently disclosed system and method, as claimed, make use of a Java

Attorney Docket No: IDF 1540 (4000-02000)

Patent

Message System (JMS). While JMS is released as an API (Application Programming Interface) for Java, not all Java systems use JMS. The use of JMS on the Java platform when communicating with other platforms has unique advantages and difficulties when implemented. These unique advantages and difficulties have not been addressed in the references because the cited reference does not use JMS on a Java platform as taught the present system.

Applicants respectfully submit that a reference does not teach or suggest receiving, interpreting, or transmitting messages sent in the JMS format simply because it uses Java. Specific support must be in place in order for a Java system to work with messages in a JMS format. This is because the API mechanisms previously used with the Java platform did not have the unique characteristics and implementation difficulties associated with JMS messaging, such as asynchronous messaging and enhanced reliability.

When Chiang et al. refers to Java, the reference is limited only to the platform on which pre-compiled computer code is executed as an application, not the format on which data of a specialized format, such as JMS, are sent and received. As previously disclosed, JMS is released as an API (Application Programming Interface) for Java. Other API mechanisms for messaging may be used with the Java, such as the GMail API for Java. This API does not support JMS messaging, yet does use the Java Platform.

Therefore, JMS, as pointed out in the Applicants' original disclosure, is a special kind of messaging system that has unique properties. Referring to Paragraph [0046] of the present application:

When the JMS network 330 needs to publish a message, a JMS application places the message in the form of a JMS MapMessage and sends the MapMessage to the JMS adapter 350. The JMS adapter 350 then converts the MapMessage into a structured event by mapping the fields in the MapMessage to the fields in the structured event. The domain name and type name properties of the message are concatenated by the JMS application in the form DomainName:TypeName and

are placed in the JMSType field in the header section of the MapMessage. The JMS adapter 350 maps the DomainName portion of the concatenated JMSType field onto the domain_name 425 field in the fixed header 422 of the structured event and maps the TypeName portion of the concatenated JMSType field onto the type_name 426 field in the fixed header 422 of the structured event. If the JMS application does not properly set the JMSType field, the JMS adapter 350 populates this field with a default value. The MapMessage does not supply a value for the event_name 427 field in the structured event. This field is instead set by the message brokering server 360 upon receipt of a message.

Chiang et al. does address the special problems associated with the JMS format. For instance, Chiang et al. does not disclose correcting errors within a JMS by populating fields with default values when there is an improperly placed value. The cited reference does not teach or suggest the use of JMS or attempt to address the unique requirements of it.

Claim 32 further recites, "communicating the message converted from JMS format to the structured event format to the middleware brokering system."

Again, the Office Action has suggested that Col. 10, lines 46-61 of Chiang et al. teaches communicating the message converted from JMS format to the structured event format to the middleware brokering system. However, Applicants are unable to find such a teaching in the cited section of Chiang et al. This section merely states that middleware 713 invokes applications 703 through the application interfaces 705. A middleware invoking the applications that it serves through their application interfaces does not teach or suggest communicating a message to a middleware brokering system as disclosed by the present application.

Claim 32 also recites, "communicating a message from a CORBA system in the structured event format to the middleware brokering system."

Again, the Office Action has suggested that Col. 10, lines 46-61 of Chiang et al. teaches communicating a message from a CORBA system in the structured event format to the

middleware brokering system. However, Applicants are unable to find such a teaching in the cited section of Chiang et al. This section merely states that middleware 713 invokes applications 703 through the application interfaces 705. A middleware invoking the applications that it serves through their application interfaces does not teach or suggest communicating a message to a middleware brokering system as disclosed by the present application.

Claim 32 also recites, "using the middleware broker to determine the destination for each of the messages from the JMS, CORBA, and mainframe systems."

Again, the Office Action has suggested that Col. 10, lines 46-61 of Chiang et al. teaches using the middleware broker to determine the destination for each of the messages from these disparate middleware systems. However, Applicants are unable to find such a teaching in the cited section of Chiang et al. This section merely states that middleware 713 invokes applications 703 through the application interfaces 705. A middleware invoking the applications that it serves through their application interfaces does not teach or suggest a middleware broker determining the destination of messages from disparate middleware systems.

Claim 32 further recites, "directing each of the messages to the appropriate one of the JMS, CORBA, and mainframe systems."

Again, the Office Action has suggested that Col. 10, lines 46-61 of Chiang et al. teaches directing each of the messages to the appropriate middleware system. However, Applicants are unable to find such a teaching in the cited section of Chiang et al. This section merely states that middleware 713 invokes applications 703 through the application interfaces 705. A middleware invoking the applications that it serves through their application interfaces does not teach or suggest directing each of the messages to the appropriate middleware system.

Furthermore, dependent Claims 33 and 34 depend directly or indirectly from allowable

Attorney Docket No: IDF 1540 (4000-02000)

Patent

independent Claim 32 and incorporate all of the limitations thereof. Accordingly, for the reasons established above, Applicants respectfully submit that Claims 32-34 are not obvious in light of the cited reference and respectfully request allowance of these claims.

Conclusion

Applicants respectfully submit that the present application is in condition for allowance for the reasons stated above. If the Examiner has any questions or comments or otherwise feels it would be helpful in expediting the application, he is encouraged to telephone the undersigned at (972) 731-2288.

The Commissioner is hereby authorized to charge payment of any further fees associated with any of the foregoing papers submitted herewith, or to credit any overpayment thereof, to Deposit Account No. 21-0765, Sprint.

Respectfully submitted,

Date: August 7, 2006

CONLEY ROSE, P.C.
5700 Granite Parkway, Suite 330
Plano, Texas 75024
(972) 731-2288
(972) 731-2289 (facsimile)



Michael W. Piper
Reg. No. 39,800

ATTORNEY FOR APPLICANT